In this lab, we will learn to use the following:

1. Implement recursive functions
2. Calculate GCD

In this lab, we are going to calculate the greatest common divisor (or GCD) of two integers, using a 2300 year old algorithm.

To compute the GCD(a,b), this algorithm first writes $a = b * q + r$ where $0 \leq r < b$ (as guaranteed by the Division Theorem). Next, repeat this by setting $a = b$ and $b = r$ until $r = 0$. Once $r = 0$, the last value for b is the GCD.

Here's the algorithm that makes it work.
function nonRecursiveGCD(a,b):
repeat until $r = 0$
    $r = a \ \% \ b$
    $a = b$
    $b = r$

Here's an example. Let's find the GCD(285, 255)

$$285 = 255 * 1 + 30$$
$$255 = 30 * 8 + 15$$
$$30 = 15 * 2 + 0$$

The final value of b was 15, so the GCD(285, 255) = 15.

It follows from the fact that if $a = b * q + r$, then GCD(a,b) = GCD(b,r). This is want makes the recursive function work so well. In the previous problem,

$$\text{GCD}(285, 255) = \text{GCD}(255, 30) = \text{GCD}(30, 15)$$

Here's another example. Find the GCD(110,42).

$$\begin{aligned}
110 = 42 * 2 + 26 &\implies \text{GCD(110,42) = GCD(42,26)}\\
42 = 26 * 1 + 16 &\implies \text{GCD(42,26) = GCD(26,16)}\\
26 = 16 * 1 + 10 &\implies \text{GCD(26,16) = GCD(16,10)}\\
16 = 10 * 1 + 6 &\implies \text{GCD(16,10) = GCD(10,6)}\\
10 = 6 * 1 + 4 &\implies \text{GCD(10,6) = GCD(6,4)}\\
6 = 4 * 1 + 2 &\implies \text{GCD(6,4) = GCD(4,2)}\\
4 = 2 * 2 + 0 &\implies \text{GCD(4,2) = 2}
\end{aligned}$$

Notice that we start with GCD(110,42), which is unknown and finish with $\text{GCD}(4, 2) = 2$, which can be used to substitute all the way back to get the answer for GCD(110,42). That's recursion!

Your assignment: Write a recursive function that calculates the GCD.
When you're done, upload your .py file to D2L.
Hint: If your code is working correctly, your code will likely be short.